Amendments to the Claims

A complete list of pending claims follows, with indicated amendments:

1. (Amended) A method for handling system management interrupts in a multiprocessor computer system, comprising the steps of:

writing a predetermined signature to a predetermined register of ~~the~~ a first processor;

executing in ~~a~~ the first processor a command of a software application to cause the first processor to initiate a system management interrupt;

receiving at each processor an instruction that ~~a software~~ the system management interrupt has been issued;

entering system management mode at each processor;

saving ~~the~~ register contents of each processor to a memory space associated with each respective processor;

selecting a second processor as ~~the~~ a system management interrupt handler, the selection of the second processor as the system management interrupt handler being accomplished according to an arbitration scheme;

scanning the contents of the memory space associated with each processor; and

when the second processor locates the saved predetermined signature in one of the memory spaces associated with the processors of the computers system, using the contents of the memory space associated with the predetermined signature for any parameters necessary for the handling of the system management interrupt.

2. (Cancelled).

3. (Cancelled).

4. (Amended) The method for issuing and handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 1, wherein the first processor and the second processor are ~~the~~ a same processor.

5. (Amended) The method for issuing and handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 1, wherein the step of executing in ~~a~~ the first processor ~~a~~ the command of ~~a~~ the software application to cause the first processor to initiate ~~a~~ the system management interrupt comprises the step of executing a software instruction causing ~~to~~ the first processor to write to a predetermined port of ~~the~~ a chip set of the computer system.

6. (Amended) The method for issuing and handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 5, wherein the predetermined port of the chip set resides in ~~the~~ a PCI bridge of the chip set.

7. (Amended) The method for issuing and handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 5, wherein the predetermined port of the chip set resides in ~~the~~ an expansion bridge of the chip set.

8. (Amended) The method for issuing and handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 7, further comprising the step of issuing from the expansion bridge ~~an~~ the instruction causing each of the processors of the system to enter system management mode.

9. (Cancelled).

10. (Cancelled).

11. (Cancelled).

12. (Cancelled).

13. (Cancelled).

14. (Cancelled).

15. (Cancelled).

16. (Amended) A method for handling system management interrupts in a multiprocessor computer system in which less than all ~~of the~~ processors of the computer system have been designated for ~~the~~ handling of software system management interrupts, comprising the steps of:

issuing an instruction from a first processor of the system to a chip set of the computer system;

receiving the instruction at the chip set of the computer system and, in response, issuing a command causing the processors of the system to enter system management mode;

writing a software system management interrupt signature to a predetermined register of the first processor as an indication that the first processor issued the command that caused the processors of the system to enter system management mode;

writing ~~the content~~ contents of the registers of each processor to a memory location, the memory location including a memory space reserved for and associated with the register contents of each processor;

selecting a second processor as the system management interrupt handler, the selection of the second processor as the system management interrupt handler being accomplished according to an arbitration scheme;

transmitting a software system management interrupt to ~~a~~ the second processor of the computer system, the second processor including a system management interrupt handler, and the second processor locating, in response, to ~~the~~ receipt of the software system management interrupt the software system management interrupt signature in the memory location; and

retrieving for use by the system management interrupt handler as parameters register contents saved by the first processor to the memory space associated with the software system management interrupt.

17. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 16 wherein the first processor ~~does not include a~~ is not operable to serve as the system management interrupt handler.

18. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 16 wherein only the second processor ~~includes a~~ is operable to serve as the system management interrupt handler.

19. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 16 wherein the instruction from the first processor to the chip set of the computer system is a write command to a predetermined port of the chip set.

20. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 19 wherein the write command is received and the software system management interrupt is issued by ~~the~~ a PCI bridge of the chip set.

21. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor computer system of claim 19 wherein the write command is received and the software system management interrupt is issued by ~~the~~ an expansion bus bridge of the chip set.

22. (Amended) A method for handling system management interrupts in a multiprocessor computer system, comprising the steps of:

receiving at one of the processors of the computer system a system management interrupt;

selecting a processor as a system management interrupt handler, the selection of the processor being accomplished according to an arbitration scheme;

scanning the memory location containing ~~the~~ saved ~~context~~ contents of each processor of the computer system;

locating in the memory location a signature identifying the saved ~~context~~ contents of the processor that issued ~~the~~ an instruction that caused the system management interrupt; and

retrieving from the saved ~~context~~ contents of the processor ~~the~~ parameters necessary for ~~the~~ handling of the system management interrupt.

23. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor system of claim 22, wherein the processor that received the system management interrupt is not the ~~same as the~~ processor that issued the instruction that caused the system management interrupt.

24. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor system of claim 23, wherein the instruction that caused the system management interrupt is a write instruction to a predetermined port in ~~the~~ a chip set of the computer system.

25. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor system of claim 24, wherein the write instruction is received at a predetermined port of ~~the~~ a PCI bridge of the computer system.

26. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor system of claim 25, wherein the write instruction is received at a predetermined port of ~~the~~ an expansion bus bridge of the computer system.

27. (Amended) The method for handling system management interrupts in ~~a~~ the multiprocessor system of claim 26,

wherein the processor that receives the system management interrupt includes a system management interrupt handler; and

wherein the processor that issued the instruction that caused the system management interrupt has not been designated for ~~the~~ handling of system management interrupts.